

Root Projection of One-Sided Time Series

Volume 96

Number 3

May-June 1991

John A. Simmons

National Institute of Standards
and Technology,
Gaithersburg, MD 20899

Until recently it has been impossible to accurately determine the roots of polynomials of high degree, even for polynomials derived from the Z transform of time series where the dynamic range of the coefficients is generally less than 100 dB. In a companion paper, two new programs for solving such polynomials were discussed and applied to signature analysis of one-sided time series [1]. We present here another technique, that of root projection (RP), together with a Gram-Schmidt method for implementing it on vectors of large dimension. This technique utilizes the roots of the Z transform of a one-sided time series to construct a weighted least squares modi-

fication of the time series whose Z transform has an appropriately modified root distribution. Such a modification can be employed in a manner which is very useful for filtering and deconvolution applications [2]. Examples given here include the use of boundary root projection for front end noise reduction and a generalization of Prony's method.

Key words: causal boundary roots; Gram-Schmidt algorithm; Prony's method; root projection; signal processing; time series; Z transforms.

Accepted: February 25, 1991

1. Introduction

Let y be a complex number. We define \hat{y} to be the "geometric sequence" vector with components $\hat{y}_n = y^n$, $n = 0, \dots, N$. Then given a time series of length $N+1$ represented by the vector $a:N$, $a(y) = \hat{y}^T a$; and the condition that y be a root of $a(y)$ is:

$$\hat{y}^T a = 0. \quad (1)$$

Given a (possibly complex) time series $a:N$ and any set of complex roots of eq (1), y_1, \dots, y_M , $M \leq N$, we shall construct a linear projection to modify $a:N$ to another time series $b:N$ so that $b(y_\alpha) = 0$, $\alpha = 1, \dots, M$, and so that

$$(a - b)^T G (a - b)^*, \quad (2)$$

where "*" means complex conjugate, is a minimum for any positive definite Hermitian weighting matrix G . In other words $b:N$ is the series closest to $a:N$ in a weighted least squares sense, which has a chosen set of complex numbers among the roots of its Y transform. We refer to this process as *root projection* (RP).

In a slightly more general context consider a (possibly complex) time series $a:N$, a set of complex numbers y_1, \dots, y_M with $M \leq N$, and a set of complex values v_1, \dots, v_M . We construct a time series $b:N$ whose difference from $a:N$ is minimal among those series with $b(y_\alpha) = v_\alpha$, $\alpha = 1, \dots, M$.

Because of the slight complication due to the complex numbers involved, we shall set and solve the above minimization problem using Lagrangian

multipliers. We want to find the time series $b:N$ such that:

$$b:N = \min_{x:N} \left[(a-x)^T G^T (a-x)^* + \sum_{\alpha=1}^M \lambda_{\alpha}^* (x^T \widehat{y}_{\alpha} - v_{\alpha}) + \sum_{\alpha=1}^M \lambda_{\alpha} (x^{*T} \widehat{y}_{\alpha}^* - v_{\alpha}^*) \right]. \quad (3)$$

Using index notation and differentiating eq (3) with respect x_j^* and λ_{α}^* , we have:

$$G_{ki} (a_i - b_i) = \lambda_{\beta} (y_{\beta}^*)^k \quad (4)$$

$$b_k (y_{\alpha})^k = v_{\alpha}.$$

Setting $y_{\alpha k} \equiv (y_{\alpha})^k$, $y_{\alpha k}^* \equiv (y_{\alpha}^*)^k$, and $G_{ij}^{-1} \equiv (G^{-1})_{ij}$; solving for b_i in the first of eqs (4), substituting into the second equation and solving for λ_{β} :

$$\lambda_{\beta} = Q_{\beta\alpha}^{-1} (a_j y_{\alpha j} - v_{\alpha}), \text{ where} \quad (5)$$

$$Q_{\alpha\beta} = y_{\alpha k} G_{kl}^{-1} y_{\beta l}^*$$

whence:

$$b_i = a_i - a_j y_{\alpha j} Q_{\beta\alpha}^{-1} y_{\beta m}^* G_{mi}^{-1} + v_{\alpha} Q_{\beta\alpha}^{-1} y_{\beta m}^* G_{mi}^{-1}. \quad (6)$$

To make the connection to linear vector space theory, we start with complex N space, \mathbb{C}^N , having an inner product given by the positive-definite Hermitian matrix G :

$$(a, b) \equiv a^T G^T b^*, \quad (7)$$

and complex M space, \mathbb{C}^M , having an inner product given by the Hermitian matrix Q^{-1} (assumed, here, to be non-singular) with an inner product definition similar to eq (7) and with vectors such as \mathbf{v} (with components v_{α} , $\alpha = 1, \dots, M$) [3]. We define the mapping Y from \mathbb{C}^N to \mathbb{C}^M by:

$$Y \equiv [Y_{\alpha i} = y_{\alpha i}; \alpha = 1, \dots, M, i = 1, \dots, N]. \quad (8)$$

Y has an adjoint mapping Y^{\dagger} , from \mathbb{C}^M to \mathbb{C}^N given by the condition $(Y^{\dagger} \mathbf{v})^T G^T \mathbf{a}^* = \mathbf{v}^T (Q^{-1})^T (Y \mathbf{a})^*$, from which

$$Y^{\dagger} = G^{-1} Y^*{}^T Q^{-1}. \quad (9)$$

In matrix notation eq (6) becomes:

$$b = (I - P) a + Y^{\dagger} \mathbf{v} \quad (10)$$

$$P = Y^{\dagger} Y, P^2 = P,$$

the projection property, $P^2 = P$, following directly from eq (6). It also follows directly from eq (6) that the range of $I - P$ is contained in the null space of Y^{\dagger} , while the range of P is contained in the range of Y^{\dagger} . Since both $I - P$ and P as well as the null space of Y and the range of Y^{\dagger} decompose \mathbb{C}^N (the latter being an orthogonal decomposition), $I - P$ and P are orthogonal mappings onto the respective spaces. The mapping $I - P$ is, then, the desired root projection.¹

2. A Gram-Schmidt Algorithm for Root Projection

In order to carry out root projection, we need an orthogonal basis for the range of Y^{\dagger} in the unitary space with metric G . From eq (8) we see that this space is spanned by vectors of the form $G^{-1} \widehat{y}_{\alpha}$. For any two such vectors:

$$(G^{-1} \widehat{y}_{\alpha})^T G^T (G^{-1} \widehat{y}_{\beta})^* = (G^{-\frac{1}{2}} \widehat{y}_{\alpha})^T (G^{-\frac{1}{2}} \widehat{y}_{\beta})^*, \quad (11)$$

where

$$(G^{-\frac{1}{2}})^2 = G^{-1}. \quad (12)$$

Thus, an orthonormal basis chosen from the vectors $G^{-\frac{1}{2}} \widehat{y}_{\alpha}$ by applying the Gram-Schmidt process using the ordinary unitary inner product will determine an orthogonal basis for the range of Y^{\dagger} in the unitary space with metric G by multiplying each vector by $G^{-\frac{1}{2}}$. However, when only a limited number of vectors need to be projected, the projection can be carried out more efficiently by:

- i) transforming the vector \mathbf{a} to $G^{\frac{1}{2}} \mathbf{a}$
- ii) projecting $G^{\frac{1}{2}} \mathbf{a}$ in the Euclidean norm using the modified basis $G^{-\frac{1}{2}} \widehat{y}_{\alpha}$
- iii) transforming the result back by multiplication with $G^{-\frac{1}{2}}$. (13)

The root projection process is especially simple when the set of y_{α} 's is closed under complex conjugation (i.e., if y_{α} belongs to the set, so does y_{α}^*) for the case of simple time or frequency weighting. Because of the closure under complex conjugation, the \widehat{y}_{α} vectors can be replaced by the vectors consisting of their real and imaginary parts, $\Re(\widehat{y}_{\alpha})$ and

¹ A concise exposition of the linear algebra required for root projection, with $G = I$ can also be given in terms of the QR decomposition [4] (although we use the "transposed" form of that in [4]).

$\Im(\hat{y}_\alpha)$. By the time weighting case we mean that G^{-1} and $G^{-\frac{1}{2}}$ are diagonal (and, therefore, positive and real), so that no unitary transformation is required. In that case each of the rows is multiplied by the same factor. Thus, the basis for the range of Y^\dagger can be obtained by applying the real form of the Gram-Schmidt process to the vectors $\Re(G^{-\frac{1}{2}}\hat{y}_\alpha)$ and $\Im(G^{-\frac{1}{2}}\hat{y}_\alpha)$. The root projection of a real time series, outlined in eq (13), will, in this case, also be real.

In the case of frequency weighting we mean scalar weights applied to the DFT components of a time series. The DFT process is, itself, a unitary transformation, and the orthogonalization process can be made real by the trick of putting $\Re(\tilde{a}_k)\sqrt{2}$ in place of the DFT component \tilde{a}_k and putting $\Im(\tilde{a}_k)\sqrt{2}$ in place of the DFT component \tilde{a}_{-k} . The real dot product of these series is the same as the unitary dot product of the DFT's. A diagonalized frequency weighting metric can then be applied to these transformed DFT series and the diagonalization process carried out. If projection is to be carried out on a real time series, it can be done in these transformed coordinates, otherwise, the new orthogonal DFT coordinates have to be reconstructed before doing the projection. Of course, the projected DFT series has to be reconverted to time series form after projection.

The real form of root projection can also be obtained by using QR algorithms such as given in LINPACK [4]. However, the implementation of the QR algorithms in LINPACK is CPU core consuming for large N and does not easily allow storage of the orthogonal matrix. These disadvantages are overcome using the Gram-Schmidt method given above. Employing a form of the Gram-Schmidt algorithm suggested by G.W. Stewart [5,6], a dynamically dimensioned Fortran 77 code was constructed allowing efficient use of a user selected buffer with the option of saving the orthogonalized basis vectors on disc for rapid repeated projection [7]. For y_α with modulus greater than one, projection was carried out using the root reciprocal to avoid overflow. In this case the series starts at y_α^{-N} and goes up to 1, a procedure equivalent to reversing the series to be projected.

3. Examples

We have principally used root projection for deconvolution. That topic is discussed in the third paper in this series [2]. Here we give four illustrative examples of root projection. The fourth example,

applying root projection to Prony's method may have some practical use. However, a detailed study has not been conducted.

1. Figure 1 shows a normalized 101 point Gaussian with 60 dB dynamic range (ratio of center to edge points). In figure 2 we show the 800 point result of convolving the first 700 points of the experimental waveform of figure 2.4a in [1] with the Gaussian of figure 1. The noise pattern shown in figure 3 results from rounding off the time series in figure 2 to 8 bit accuracy. The rounded-off time series is not shown. The standard deviation of this noise is 6.02×10^{-4} .

A basis for all 800 point time series can be formed from the waveform of figure 2 together with the 799 geometric root vectors formed from the roots of the Y transform of that waveform. The noise vector of figure 3 must then be a linear combination of all these vectors, while the true signal

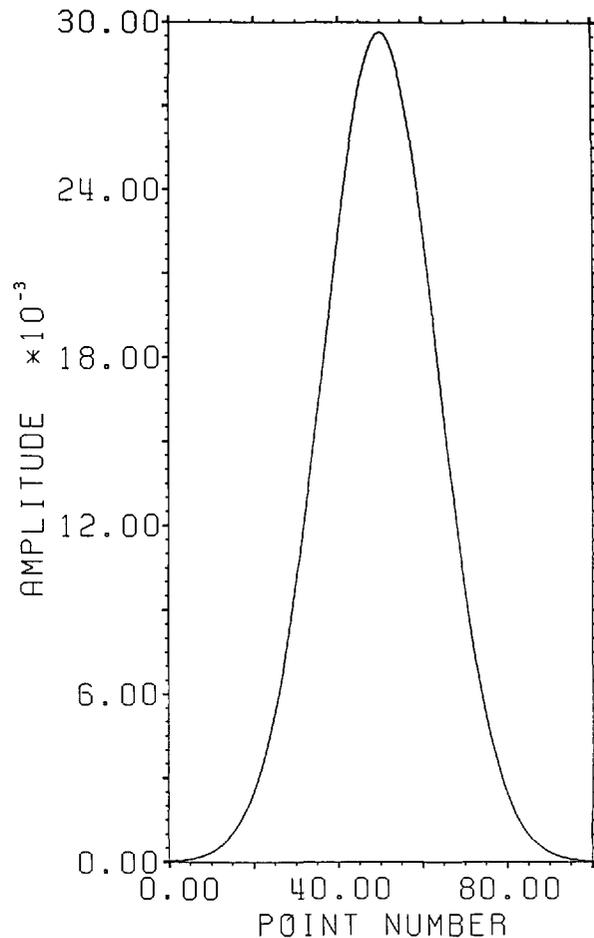


Figure 1. Normalized 101 point Gaussian with 60 dB dynamic range.

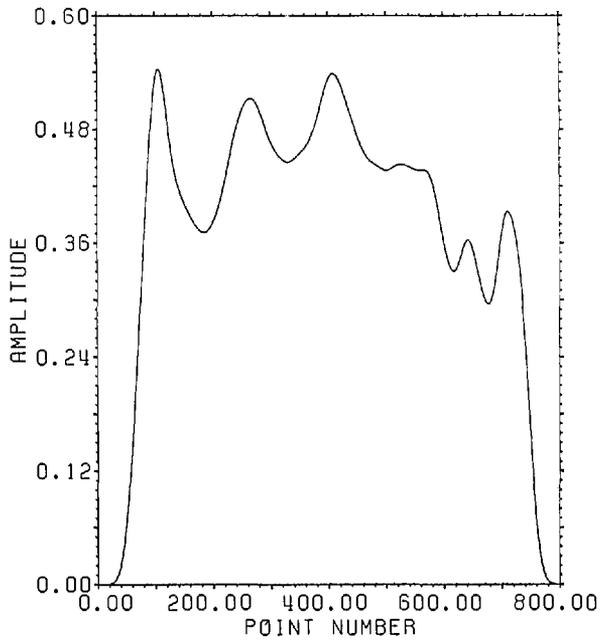


Figure 2. Convolution of the first 700 points of the time series in figure 2.4a of reference [1] with the 101 point time series of figure 1.

vector of figure 2, which is one of the basis elements, is orthogonal to all of the 799 geometric root vectors. If we use the geometric root vectors built from the Gaussian of figure 1 and apply root projection to the rounded-off series built from the series in figure 3, the noise of the resultant series will be reduced in magnitude, since it is orthogonal to the 100 geometric root vectors formed from the roots of the Y transform of the Gaussian. The standard deviation of that noise is 5.6×10^{-4} , almost exactly the theoretical estimate of $(\frac{7}{8})^{\frac{1}{2}}$ expected for Gaussian noise. If the 799 geometric root vectors from both the Y transforms of the Gaussian and the waveform of figure 2.4a in [1] are used for the projection, then the error appears as in figure 4—a mini-image of the correct time series—with a standard deviation of 2.0×10^{-5} (again approximately $(800)^{-\frac{1}{2}}$ of the original noise deviation). This time, of course, the noise is biased with a mean value of 4.9×10^{-5} . This curve also indicates the precision of the root finding and Gram-Schmidt routines used in these calculations.

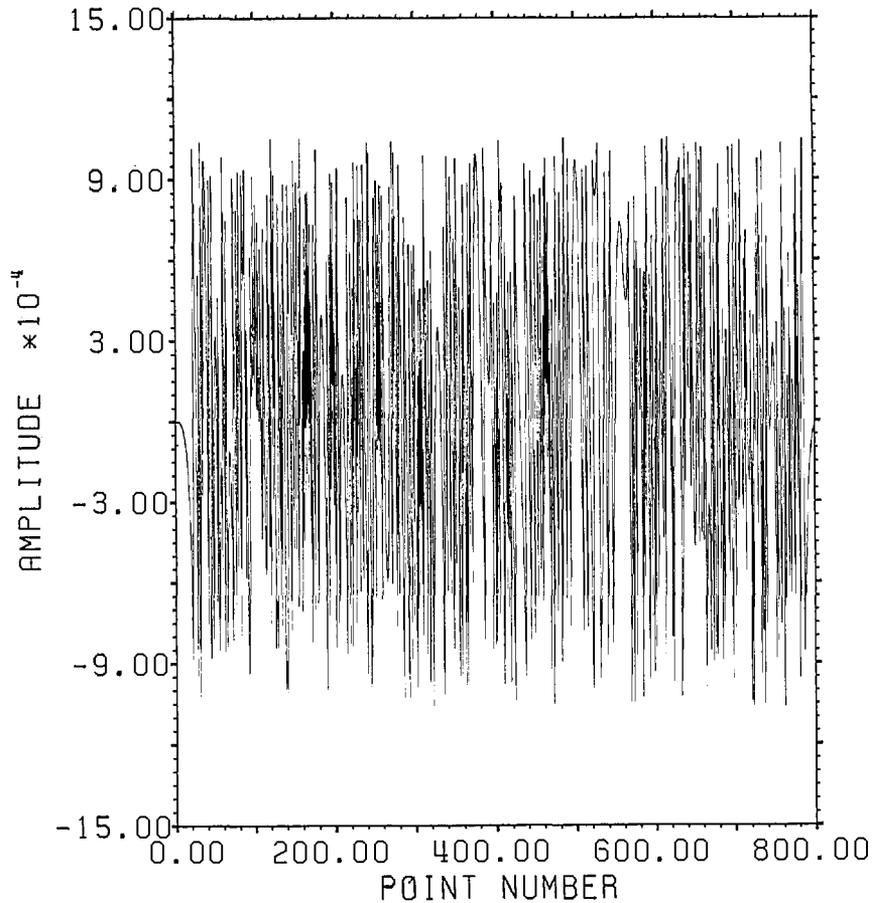


Figure 3. 8-bit roundoff noise for the series in figure 2.

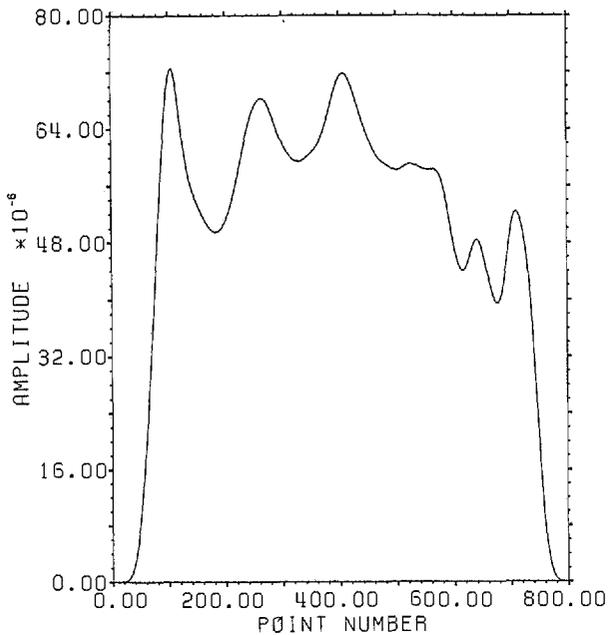


Figure 4. Error from projecting the series in figure 2 plus noise through all roots of the Y transform of the series in figure 2. Note that the remaining error is proportional to the original series.

2. The use of an extensive causal boundary root set for front end filtering is shown in figures 5a and 5b. To the Gaussian filtered curve of figure 2.3a in [1] we add the -40 dB noise distribution shown in figure 5a and apply root filtering using only the causal boundary roots from figure 2.1b of [1]. The noise after filtering showing extensive front end reduction is shown in figure 5b.

3. A more general example of root projection employing both time and frequency weighting is afforded by constructing an approximation to an optimal maximal-ripple lowpass filter. Figure 6 shows such a filter of 151 elements designed using the Remez exchange algorithm of McClellan et al. [8]. In this case the passband was set for 20% of the Nyquist frequency and the stopband for frequencies over 33% of the Nyquist frequency.

We start with a delta series shifted so that the value 1.0 lies at position #76 in the center of the time interval. To use unweighted projection and 101 roots evenly spaced from -60° to $+60^\circ$ on the unit circle would produce the familiar series obtained

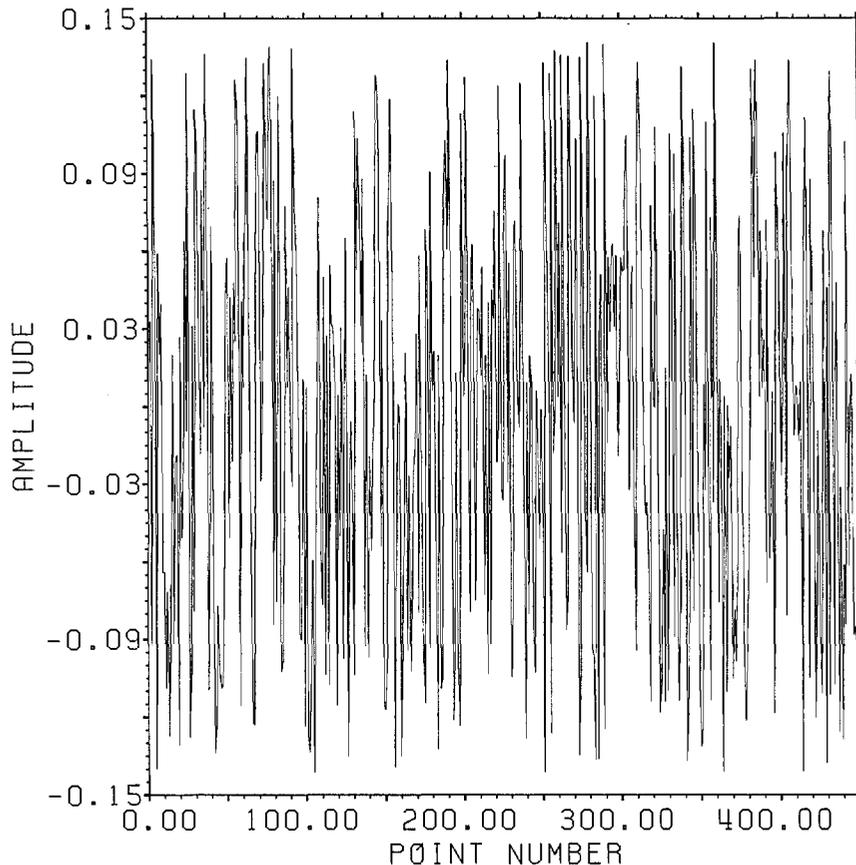


Figure 5a. Noise added to series in figure 2.3a of reference [1].

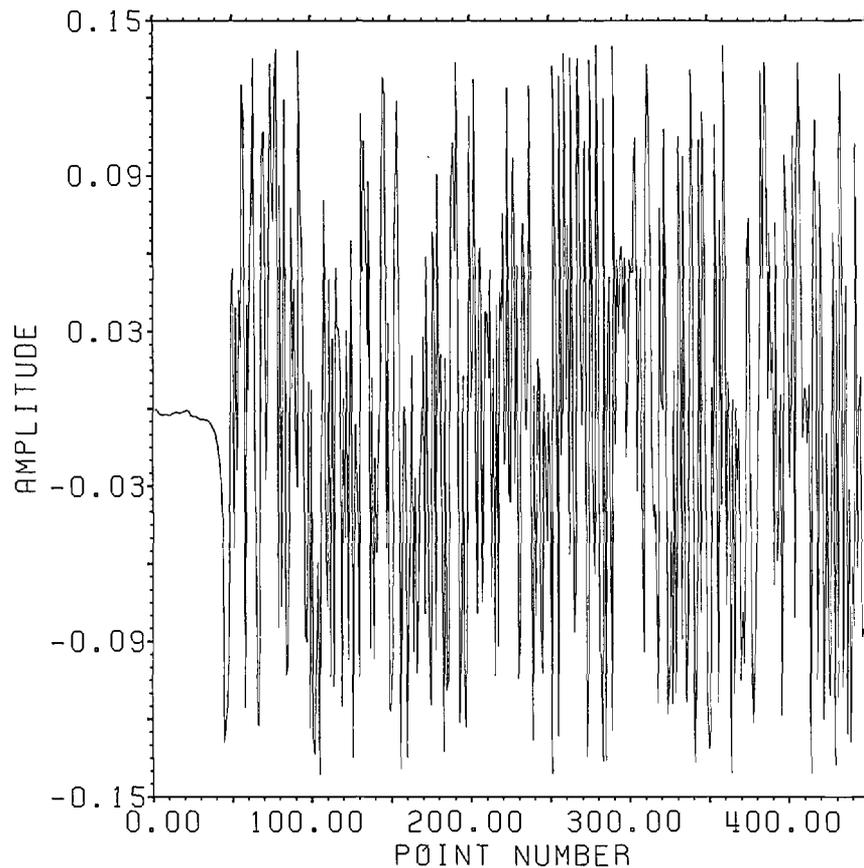


Figure 5b. Noise after filtering using causal boundary roots of the series in figure 2.3a of reference [1] showing extensive front end noise reduction.

from windowing the DFT of the shifted delta series. However, by using time weighted projection with the center symmetric time weight shown in figure 7 and adding two extra roots near each edge of the stopband region, one can produce a filter with 120 dB attenuation in the stopband and $\sim 2.5 \times 10^{-7}$ leakage in the time domain, but with 0.11 ripple in the passband. Without time weighting the filter produced has 64 dB attenuation in the stop band and 0.08 ripple in the passband, but with $\sim 1.0 \times 10^{-3}$ leakage in the time domain.

To remove the passband ripple from the weighted filter we subtract from it the shifted delta series. The resulting series should be close to zero in the passband region. We place 31 roots evenly in the passband and again place two extra pairs of roots near the edge of the band. Applying frequency weighted projection with a simple square frequency weight to hold the stopband attenuation in place, we obtain (after re-adding in the shifted delta series) the doubly root projected filter shown in figure 8, where it is compared with the unweighted doubly projected and optimal filters. The

weighted series closely resembles the optimal series of figure 6 with leakage at the ends of the time interval of $\sim 8.0 \times 10^{-5}$ as opposed to $\sim 2.0 \times 10^{-3}$ for the unweighted filter. The leakage for the optimal filter is $\sim 6.0 \times 10^{-8}$. The attenuation in the stopband is 70 dB and the ripple in the passband is 1.0×10^{-4} for the weighted filter while the unweighted filter has 36 dB attenuation and 8.0×10^{-4} ripple. The optimal filter has 116 dB attenuation and 5.0×10^{-5} ripple. The spectra for the three filters are compared in figure 9.

The root patterns in the stopband for the optimal filter are shown in figure 10. As can be seen, they also cluster near the edge of the stopband, but they are not evenly spaced throughout the stopband. Similarly, the roots of the optimal filter minus the shifted delta series, shown in figure 11, exhibit the same uneven spacing and clustering in the passband region.

4. Root projection can be used to provide a "global" least squares generalization to the Prony method. Prony's method is applied to the case of

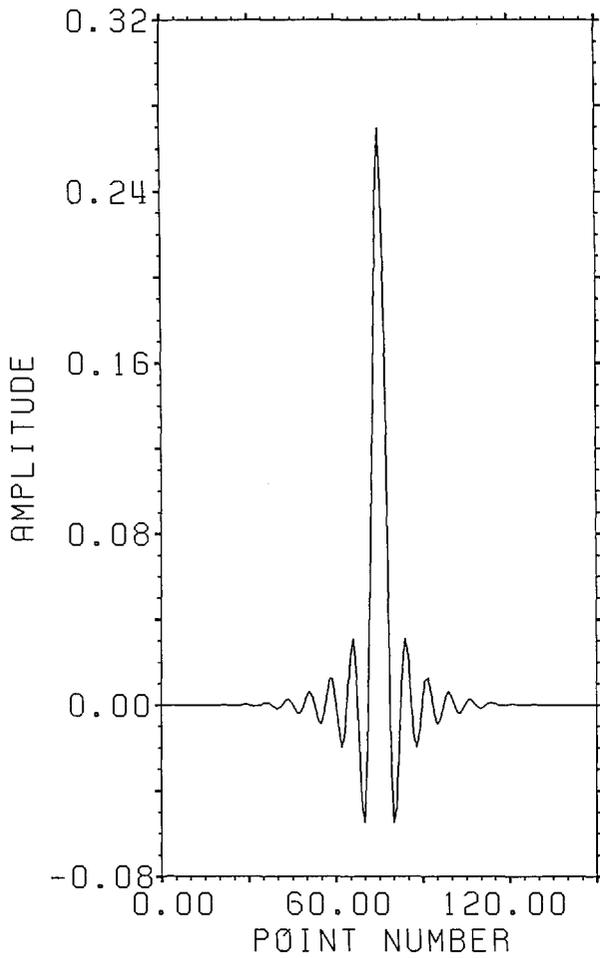


Figure 6. Optimal maximal ripple filter of 151 elements.

$N + 1$ data points which one wants to represent as the sum of n decaying exponentials (Prony takes $N + 1$ even and $n = (N + 1)/2$):

$$F_k = \sum_{j=1}^n A_j y_j^k, \quad k = 0, \dots, N. \quad (14)$$

Here y_j is thought of as a complex number of modulus less than 1, $y_j = e^{\beta_j T}$, $\text{real}(\beta_j) < 0$, and T a sampling interval (See, e.g., [9]). Eq (14) can be rewritten, taking advantage of the cyclotomic polynomial expression, as

$$F(y) = \sum_{k=0}^N F_k y^k = \sum_{j=1}^n A_j \frac{1 - (y_j y)^{N+1}}{1 - y_j y}. \quad (15)$$

Combining the terms in eq (15) together to form a single fraction gives

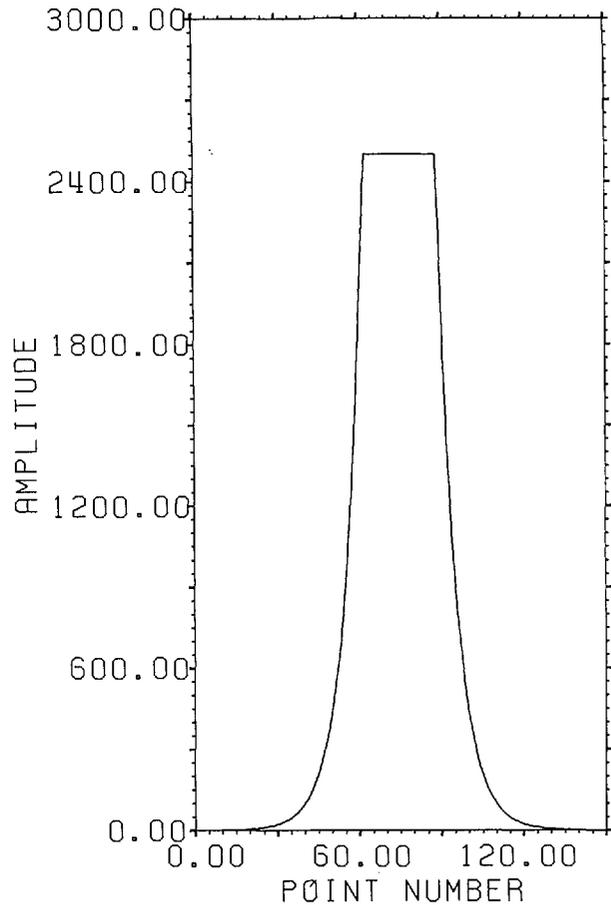


Figure 7. Time weights for approximating an optimal filter.

$$F(y) = \frac{[A_1(1 - y_1 y) \dots (1 - y_n y) (1 - (y_1 y)^{N+1}) + \dots + A_n(1 - y_1 y) \dots (1 - y_n y) (1 - (y_n y)^{N+1})]}{[(1 - y_1 y) \dots (1 - y_n y)]} \quad (16)$$

Equation 16, then, has the form

$$F(y) = \frac{P(y) + y^{N+1} R(y)}{Q(y)}, \quad (17)$$

where $Q(y)$ is a polynomial of degree n whose roots are $(y_j)^{-1}$, and $P(y) + y^{N+1} R(y)$ is a polynomial of degree $N + n$ whose $N + 1 - n$ coefficients from that of y^n to that of y^N are zero. The roots of $Q(y)$ are referred to as the poles of F . Conversely one can show that if eq (17) holds with the corresponding coefficients zero, then setting

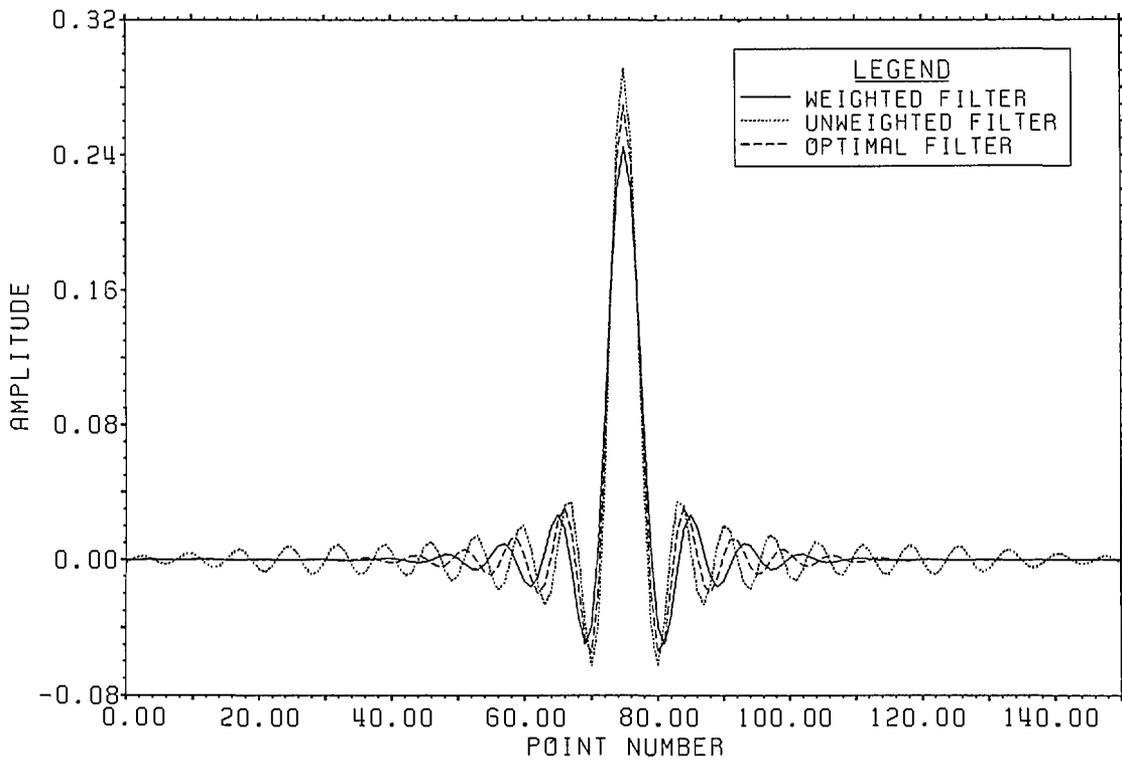


Figure 8. Comparative time plot showing the effects of using weighted projection for approximating an optimal filter.

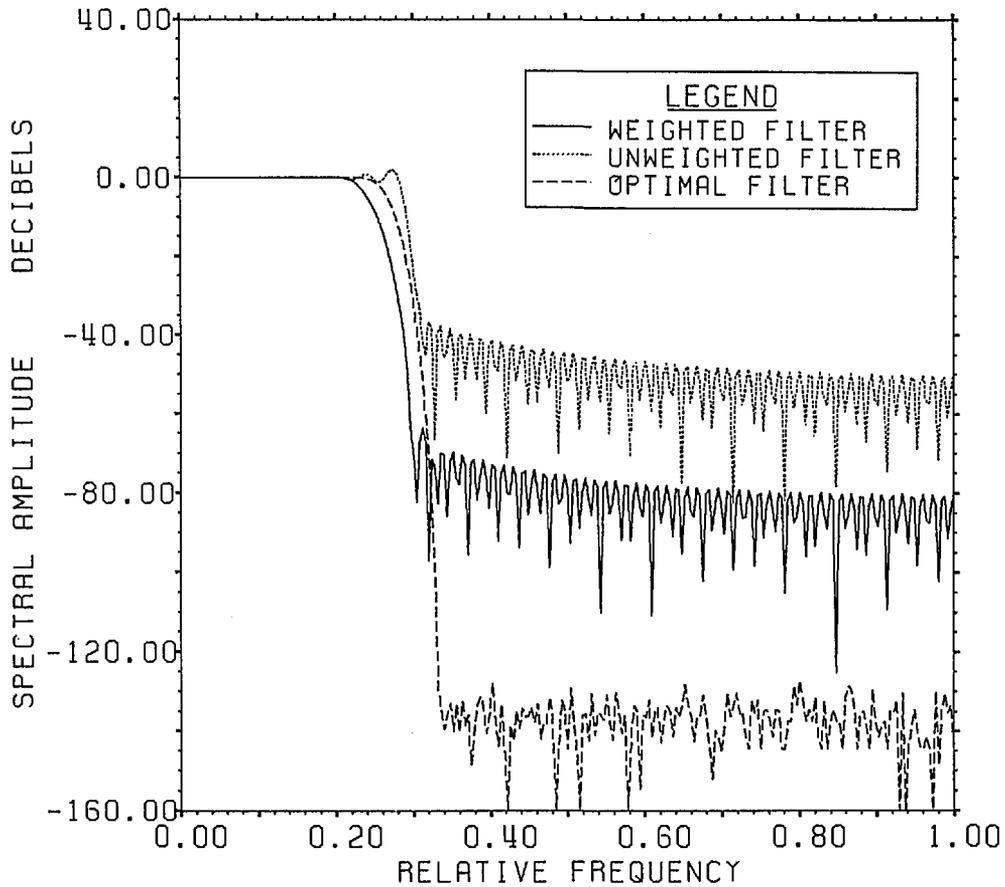


Figure 9. Comparative frequency plot showing the effects of using weighted projection for approximating an optimal filter.

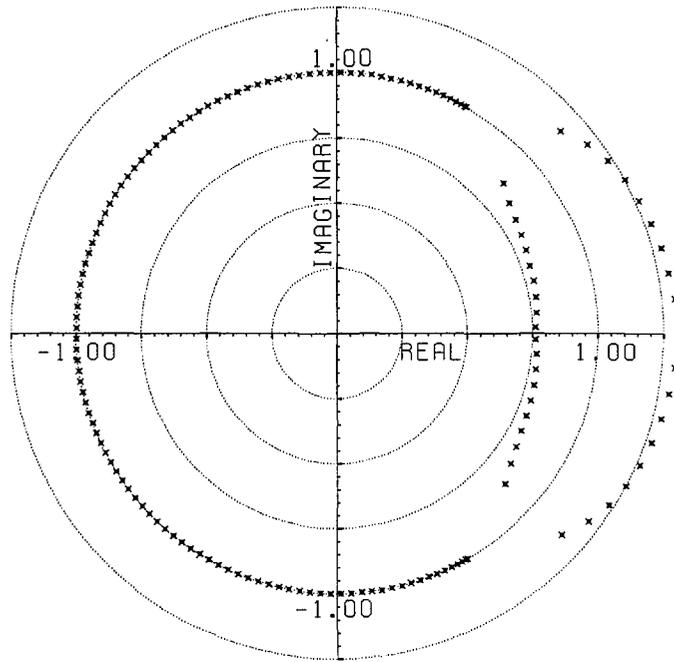


Figure 10. Roots of the Y transform of the optimal filter shown in figure 6 showing roots in the stopband.

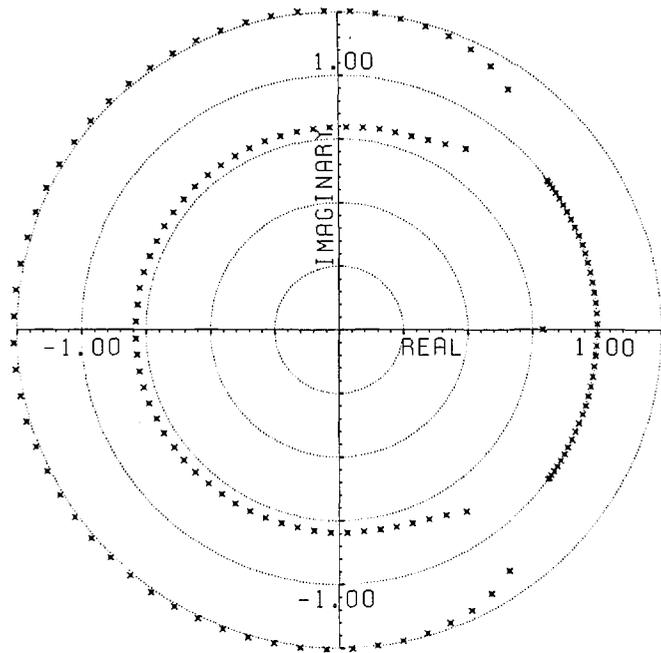


Figure 11. Roots of the Y transform of the optimal filter of figure 6 minus the shifted delta series showing roots in the pass band.

$$Q(y) = \prod_{j=1}^n (1 - y_j y) \quad (18)$$

$$A_j = -\frac{P(1/y_j)}{Q'(1/y_j)}$$

yields the expression 14.

The goal, then, is to solve the algebraic equation:

$$Q(y)F(y) = P(y) + y^{N+1}R(y), \quad (19)$$

where P , Q , and R are all unknown. Only the integer n is given. This is generally a very ill-conditioned process, but when the y_j 's are real or occur in conjugate pairs, the following root projection procedure is often successful:

a. Use an initial estimate for the coefficients, c_i , of $P + y^{N+1}R$ by setting $c_i = \kappa_1$ for $i = 0, \dots, n-1$, $c_i = 0$ for $i = n, \dots, N$, and $c_i = \kappa_2$ for $i = N+1, \dots, N+n$; and use for the diagonal weighting matrix, W , the series w_l with $w_l = \lambda_1$ for $l = 1, \dots, n$, $w_l = 1$ for $l = n+1, \dots, N+1$, and $w_l = \lambda_2$ for $l = N+1, \dots, N+n+1$, where κ_1 , κ_2 , λ_1 , and λ_2 are user selected values.

b. Project the series made from $P + y^{N+1}R$ into the range of F using W and the geometric row vectors made from the roots of $F(y)$. The projected polynomial will have negligible values for the coefficients between $n+1$ and $N+1$ (depending on the noise in the data) and will be divisible by $F(y)$. Further, $P(y)$ can be read off from the projection and $Q(y)$ found by simple division (using FFT methods, for instance, as discussed in [2]).

As long as: *i*) the projection is stable, *ii*) the projected vector is not zero, and *iii*) the values of the weights are large enough to produce an answer within the noise for similar cases with no data noise, then the values of κ_1 , κ_2 , λ_1 , and λ_2 have very little effect on the calculated y_j 's and A_j 's. In the ordinary Prony case there are as many degrees of freedom added in R as are restricted between P and R (one degree is an irrelevant multiplicative constant since F is expressed as a ratio).

This method has been tried for several examples with both real and complex poles with maximum to minimum root amplitude ratios up to 60:1 and with minimum root separation down to 0.01. Using double precision data (with $\kappa_1 = 10^8$, $\kappa_2 = 10^4$, $\lambda_1 = 10^{28}$, and $\lambda_2 = 10^{18}$) it works well up to about eight poles after which the positions of the larger poles start to degenerate (due to the ill-conditioning of the prob-

lem). The representation of the time series coefficients of F remains accurate to a relative error of about 10^{-12} , which represents approximately the cumulative accuracy of the algorithms involved. Using data given to 16 bit precision (around 5 place accuracy), the method is able to resolve about 3 poles (where $\kappa_1 = 10^3$, $\kappa_2 = 10^2$, $\lambda_1 = 10^{12}$, and $\lambda_2 = 10^6$) with the positions of the larger poles again beginning to degenerate first.

4. Summary

The idea of root projection (RP) was introduced to permit least-squares modification of a one-sided time series allowing a set of given complex numbers to be roots of the Y transform of the time series. The general framework of the method was presented, and techniques were given to adapt the method to the Gram-Schmidt algorithm. For time and frequency least-squares weighting, a dynamically dimensioned form of the Gram-Schmidt algorithm has been developed to carry out root projection. The code has been employed for root projection on time series with up to 1600 points and achieved better than 12 place accuracy. Illustrative examples of root projection were shown for noise reduction and filter construction as well as a least-squares extension of Prony's method.

Acknowledgments

The author wishes to thank D. P. O'Leary and G. W. Stewart for many valuable numerical suggestions and comments. This research was carried out under a joint NBS/DARPA program on NDE/RSP.

5. References

- [1] Simmons, J. A., "Boundary Roots and Signature Analysis of Causal and Transient Time Series," submitted to SIAM J. on App. Math.
- [2] Simmons, J. A., J. Res. Natl. Inst. Stand. Technol. **96**, 345 (1991).
- [3] Halmos, P. R., Finite-Dimensional Vector Spaces, Springer Verlag (1974).
- [4] Dongarra, J. J., Moler, C. B., Bunch, J. R., and Stewart, G. W., LINPACK User's Guide (2nd edition), Chapter 9, SIAM, 1981.
- [5] Stewart, G. W., private communication.
- [6] Rutishauser, H., Numerische Mathematik **16**, 205 (1970).
- [7] Simmons, J. A., unpublished (Fortran '77 code available on request).
- [8] McClellan, J. H., Parks, T. W., and Rabiner, L. R., IEEE Trans. Audio Electroacoust. **AU-21**, 506 (1971).

- [9] Twomey, S., Introduction to the mathematics of inversion in remote sensing and indirect measurements, Elsevier/North Holland Press (1977) pp. 172-174.

About the author: John A. Simmons is a mathematician in the Metallurgy Division of the Materials Science and Engineering Laboratory of NIST, and has worked in the elasticity and elastodynamics of defects in solids. This work has arisen from studies of the signal processing of acoustic emission in materials.